particular destination. If there is no such entry in the FD, the bridge forwards the frame through all ports except the port from which the frame originates. Whenever a frame from source *s* arrives from port *p*, the bridge marks in its FD that the forwarding port of *s* is *p*. As the learning process is simple, if there are loops in the bridged LAN, a frame may be forwarded indefinitely. To avoid this undesirable feature, function (3) mentioned above is used to make sure the active topology among the bridges is always a tree so that there is a unique path between each pair of bridges. We refer to such a path herein as a tree path.

The spanning tree algorithm builds a unique shortest path tree rooted at the root bridge in a distributed manner. This root bridge is selected using bridge identifiers. A path connecting the root bridge and another bridge over the spanning tree is referred to as a root path associated with the bridge. By exchanging configuration messages, bridges identify the root bridge and select which ports to activate. For each LAN, a single bridge is elected among all bridges connected to the LAN to be the *designated bridge*, such that it is the bridge that is closest to the root bridge. In order to maintain an up-to-date tree that reflects the underlying topology, the root bridge broadcasts configuration messages periodically over the spanning tree to all other bridges, for example, approximately every four (4) seconds.

The IEEE 802.1D standard defines two types of Bridge Protocol Data Unit (BPDU), namely Configuration BPDU and Topology Change Notification BPDU. Bridges send MAC frames containing Configuration BPDU to each other in order to communicate topology information and compute the spanning tree. Bridges send MAC frames containing Topology Change Notification BPDU up the spanning tree to inform the root bridge of a topology change. Each Configuration BPDU MAC frame includes a MAC header that contains a source MAC address and a destination MAC address. The source MAC address is the MAC address on the port of the bridge originating the Configuration BPDU MAC frame. The destination MAC address field carries the bridge group MAC address so that the Configuration BPDU MAC frame is received by all the bridges in the extended LAN. The information in the Configuration BPDU may be used by a bridge in preparing its own Configuration BPDU MAC frame. Each Configuration BPDU contains a BPDU Header and a set of BPDU Parameters. The BPDU Header consists of a Protocol Identifier field, a Protocol Version Identifier field and a BPDU Type field. The Protocol Identifier takes a specific value that identifies the Spanning Tree Bridge Protocol. The Topology Change Notification BPDU consists merely of a Protocol Identifier field, a Protocol Version Identifier field, and a BPDU Type field with a code reserved for this type. When a bridge detects a change in the active topology of the spanning tree, it sends a Topology Change Notification BPDU to the root bridge. The root bridge will then broadcast it to all bridges in the extended LAN. The encoding for the fields in the Configuration BPDU and the Topology Change Notification BPDU can be found in the IEEE standards.

In order to balance traffic load, extensions to the IEEE 802.1D Spanning Tree Bridge Protocol have been proposed to allow non-tree links to be used for frame forwarding under appropriate conditions. These extensions consider alternate paths that traverse at least one non-tree link.

For example, U.S. Patent Number 4,811,337 (Hart) discloses a method, known as *distributed load sharing (DLS)*, to allow non-tree links to be selected for frame forwarding. In accordance with the method of Hart, a forwarding path is either a tree path, a DLS link, or a DLS path that is a concatenation of DLS links. This method requires each selected DLS link to satisfy the following conditions:

a) The two DLS bridges at the ends of the selected DLS link must both implement DLS.

b) The two bridges at the ends of the selected DLS link must be such that one of them is not the ancestor of the other in the spanning tree.

c) The length associated with the selected DLS link must be no greater than the sum of the root path distances associated with the two DLS bridges.

Condition (b) above is necessary in order to prevent any intermediate bridge on the tree path between the two DLS bridges to misinterpret a forwarding direction of a particular end station. In view of condition (c), Hart's approach can overestimate the actual length of a tree path between two DLS bridges. In this respect, a non-tree link between a pair of DLS bridges may actually be selected even though it has a greater length than the length of the corresponding tree path. There is no such problem only when the root bridge is on the tree path between the two DLS bridges. Thus, this method cannot guarantee that a forwarding path is no worse than its corresponding tree path for any additive metric considered.

U.S. Patent Number 5,150,360 (Perlman, et al.), extended Hart's DLS method to address certain shortcoming of the DLS method. Specifically, Perlman, et al., proposed to identify non-tree links so that they can be used for forwarding frames without traveling a long way on the spanning tree. The approach is simpler than DLS and can utilize any non-tree link connecting a pair of bridges that have implemented the extended protocol, referred to as Generalized DLS (GDLS). GDLS does not select a non-tree link between a pair of GDLS bridges to be a GDLS link by comparing the length of the non-tree link to that of the corresponding tree path. Instead, GDLS compares the "speed" of the non-tree link to that of the corresponding tree path. The "speed" of the non-tree link and that of the tree path are determined by having one of the GDLS bridges send to the other GDLS bridge a special protocol data unit over the non-tree link and another over the tree path. Separate information has to be kept for every non-tree link even though some links will not be used at all. The method of Perlman, et al., cannot guarantee that a forwarding path is no worse than its corresponding tree path for any additive metric considered except when the additive metric is delay. Incidentally, this method is backward compatible with the IEEE 802.1D Spanning Tree Bridge Protocol.

Another prior art method dynamically creates a shortest path tree rooted at a given source starting with a default spanning tree. Some non-tree links are activated and some tree links are disabled on demand according to a delay measure. Information kept in bridges grows quadratically with the number of ports in the bridges. This method is not backward compatible with respect to the IEEE 802.1D Spanning Tree Bridge Protocol.

A bridge learning protocol has also been devised so that optimal or suboptimal routes can be identified. Cost to each known end station is kept for each port. The protocol works similarly to the distance vector method and is backward compatible with respect to the IEEE 802.1D Spanning Tree Bridge Protocol. Topology information is kept by every bridge for every port. Moreover, when there are bridges that do not execute the protocol, a path found by the protocol may be longer than its corresponding tree path.

Prior art methods also propose to maintain distance vectors in bridges showing the shortest path direction for getting to a particular LAN, and not to a station. *Mapping tables* are used to map stations to LANs. When a frame is received, the bridge maps the target station to the target LAN and finds the forwarding port from the distance vector. Mapping tables are exchanged by means of flooding (standard for distributing local information throughout the network). This protocol is not backward compatible with respect to the IEEE 802.1D Spanning Tree Bridge Protocol. There are bridge architectures that have IP routing features. Bridges exchange topology information to obtain the complete topology of the extended LAN. Once the complete topology is synchronized, the shortest path to every LAN can be found. Their architecture also has a mechanism to locate end station to LAN, which is similar to the mapping tables. These protocols are not backward compatible with respect to the IEEE 802.1D Spanning Tree Bridge Protocol.

SUMMARY OF THE INVENTION

The present invention is a novel bridge protocol, which will be referred to hereafter as a Spanning Tree Alternate Routing (STAR) Bridge Protocol, that has important advantages over the present day techniques discussed above. The proposed bridge protocol of the present invention is transparent to end stations, backward compatible with the current IEEE 802.1D standard, free of loops in frame forwarding, and easy to implement with scalable message overhead and storage requirement. In the protocol disclosed herein, forwarding paths are selected based on a path metric that is a sum of link metric values of all the links along the path, wherein the link metric is any desirable cumulative metric, such as delay, administrative cost, number of hops, etc. We will show that our protocol always finds a path that is either shorter or the same as the path on the spanning tree. This protocol is also backward compatible with the IEEE 802.1D standard, such that existing bridges need not be modified and new bridges, henceforth referred to as STAR bridges, operate seamlessly with the existing standard. The STAR Bridge Protocol attempts to find and forward frames over alternate paths that are shorter than their corresponding tree paths on the standard spanning tree, and makes use of the standard spanning tree for default forwarding. We refer to the shorter alternate paths herein as enhanced forwarding paths. The protocol may use any of a variety of distance metrics for evaluating forwarding paths, including number of hops, physical length, transmission delay, and cost, provided the metric used is supported by the standard bridge protocol. In one embodiment of the present invention, all frames sent from a source bridge to a destination bridge are forwarded over either a standard tree path or an enhanced forwarding path, but not both. In another embodiment of the present invention, all frames sent from a source bridge to a destination bridge are forwarded over standard tree paths by default,

6